

1 Objectives

You can often tell what the language designers thought about their language by the libraries that are included with it. Many of Prolog's involve the analysis of structures. In this lecture we will go over some of the builtin predicates of Prolog.

- Know how to use `call` and `assert`
- Know how to use `findall` and `checklist`.

2 Examples

```
1 ?- dynamic likes/2.
2 ?- likes(john,mary).
3 No
4 ?- assert(likes(X,Y) :- likes(Y,X)).
5 ?- assert(likes(john,mary)).
6 ?- likes(mary,X).
7 ERROR: Out of local stack
8 ?- retract(likes(john,mary)).
9 Yes
10 ?- asserta(likes(john,mary)).
11 Yes
12 ?- likes(mary,X).
13 X = john
```

3 Problems

Try the following problems. In a few minutes the instructor will go over the solutions. Feel free to work with the person next to you!

1. Write a function “**says**” that takes two arguments. The first is the name of a person making the implication. The second is a prolog expression. Record the claim, and then tell the prolog.
2. Next, suppose we can find out later that some people aren't reliable, and we should no longer believe anything they say. Write a function “**disbelieve**” that takes the name of a person and retracts everything they said before.

```
1 ?- says(frank,likes(john,mary)).
2 ?- likes(john, mary).
3 Yes
4 ?- disbelieve(frank).
5 ?- likes(john, mary).
6 No
```